4. Data Acquisition

Detectors for the experiment are located in the pit of the AP-50 target hall at Fermilab. As events are selected by hardware triggers located near the interaction vertex of the hydrogen gas jet and the antiproton beam(which determine whether the event should be kept or passed over), the respective pulse heights from various detectors are sent upstairs through delay cables to the AP-50 Counting Room. Further decisions are made via software regarding which events are worth keeping and written to tape and/or disk.

The data are often referred to by their run number. Generally a "run" is the running of the data acquisition system and the writing of event information to tape. A new run begins when one of the many tapes becomes full of data and a new set of tapes must be used. Each run has its own set of calibration constants in the E835 database.

The basic philosophy of the E760/E835 trigger is to base decisions on a more manageable number of components than the 1280 elements of the CCAL while at the same time requiring at least two significant deposits of transverse electromagnetic energy (i.e. hopefully a e^+e^- or $\gamma\gamma$ signal). This is achieved by analog summing adjacent calorimeter blocks into 40 *super-clusters.* The summing process includes overlaps so that most of the energy from the transverse spread of a shower may be contained within a single super-cluster, and higher trigger rates are avoided by setting proper thresholds[47].

4.1. The Trigger

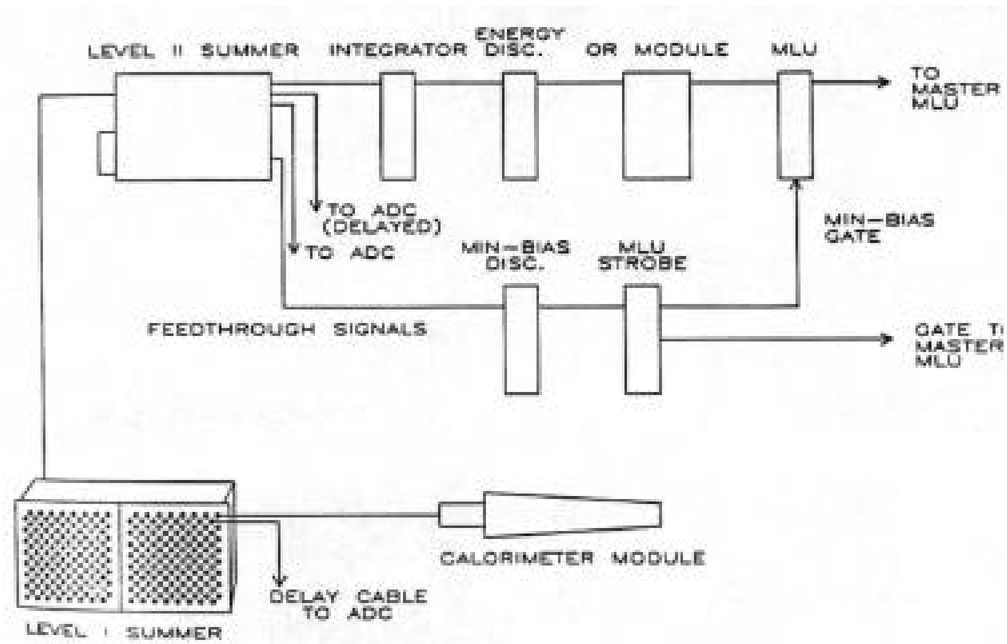

Figure 4.1 : The CCAL Trigger Logic [47].

Initially, the signal from each of the 1280 blocks is split: 95 % is sent through a delay cable to the ADC's , and 5 % is delivered to the Level I Summer (Figure 4.1). Ultimately if the event passes certain requirements set on these super-clusters and/or the inner detectors, a gate signal will be sent by the Gatemaster(an event must pass the requirements of at least one hard-wired trigger in this logic unit) to the FERA's, allowing them to read in the analog signals from the previously mentioned delay cables. Otherwise the event is rejected.

Signals within each wedge of 64 blocks are summed into one of 8 *super-wedges*, i.e. a super-wedge is a pure azimuthal sum. Each super-wedge is actually a sum of 9 blocks due to the overlap mentioned earlier. Hence the first and last blocks in a ring are included in two different sums. Since there are 20 such wedges, the Level I Summer produces 20 X 8 = 160 signals.

These 160 Level I signals are sent to the Level II Summer, which sums all Level I signals sharing one of the 8 azimuthal regions into 5 polar regions called *super-rings* , which overlap with the adjacent super-ring by one calorimeter block. Since there are 8 azimuthal regions and 5 sums performed in each region, there are 8 X 5 = 40 *super-clusters*.

The requirement of having at least two significant deposits of transverse electromagnetic energy translates into requesting two such deposits to be a) coplanar and b) have an invariant mass of greater than 2.0 GeV. Usually the coplanar requirement means that the super-clusters must be back-to-back in the center-of-mass frame, which means that the other energy deposit must lie in the directly opposite super-cluster(called a "1 vs. 1" geometry). However, this can be softened to a 1 vs. 3 geometry: The other energy deposit is in one of the 3 opposing super-clusters in the center-of-mass.

This constitutes the main body of the trigger. Other requirements, such as using or vetoing on different detectors or changing the invariant mass threshold, may be added to produce different triggers. Several different hardware triggers are flagged as on or off by the Gatemaster, including a e^+e^- trigger, $\pi^+\pi^-$ trigger, $\gamma\gamma$ trigger, $\phi\phi$ trigger, and several calibration triggers.

4.2. DART Hardware

To get events flagged by the Gatemaster written to tape, they must pass through the hardware of the E835 data acquisition system (Figure 4.2). The E835 DAQ is based on the architecture of the DART Data Acquisition Project shared among several fixed target experiments at Fermilab. DART provides a common system of hardware and software that can be easily configured across a variety of UNIX and VME platforms[48].

In regards to hardware, the E835 DAQ uses three Silicon Graphics computers and one Motorola MVME167 processor. One Silicon Graphics Indy runs the programs that perform Run Control, executes the Data Acquisition Monitoring Program (DAMP), and talks to the two CAMAC branches. A Challenge-L takes advantage of four 150-MHz processors to build and filter events, and then log them to tape. An Indigo is used for monitoring the detectors and runs the Event Display, in which signals for all the detector elements for a chosen event may be viewed graphically[49].

Data from the detectors are read out via TDC (LRS3377)[50] in a common start mode and either by FERA ADC (LRS4300)[51] or PCOS Latches (LRS2731)[52] through the ECL-ports in the controller’s front by Damn Yankee Controllers (DYC)[53] . The TDCs (time-to-digital converters) help determine if a particular hit in the detector is on-time or not associated with this event. ADCs (analog-to-digital converters) convert the analog signals delivered by the detectors' photomultiplier tubes into digital information for further processing. When properly calibrated, the ADCs yield information about the energy of a hit in the detector for example. A data stream is formed by daisy-chaining DYC’s together by a DART ribbon cable form streams, and each stream is read out by two pairs of Access Dynamics DC2/DM115 modules[54].

The DC2 writes its data to one of the Dual Ported Memories (DPM) using a “ping-pong” algorithm. A process on the Challenge called Gateway reads the “ping” memory while the DC2 writes to the “pong” memory. A third DPM serves as a mailbox which is constantly polled by the processes that write and read to the first two Dual Ported Memories to see when the other is finished[55].

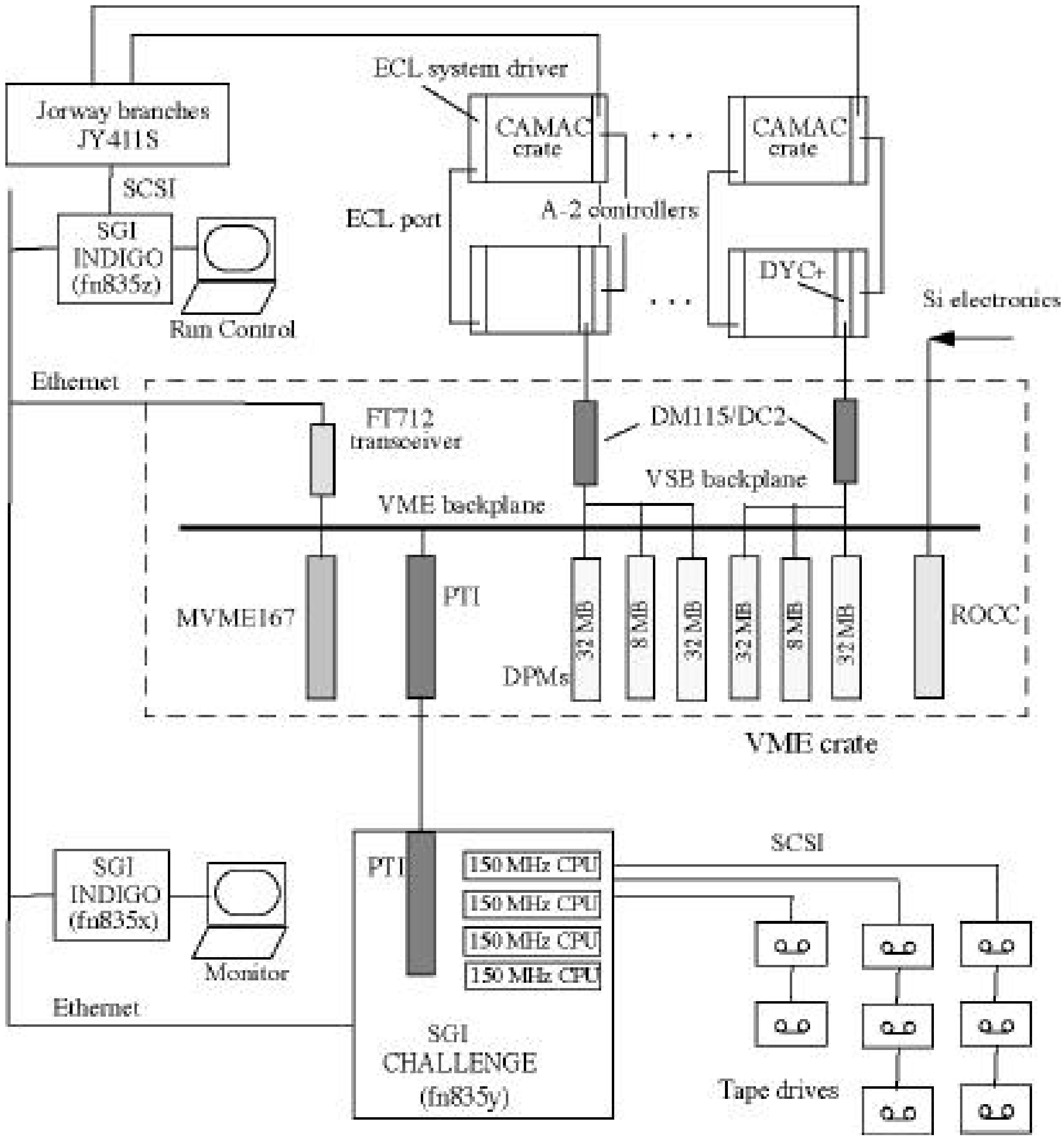

Figure 4.2 : Hardware of the E835 Data Acquisition System [56].

The event header[57] , which in effect lists the locations of different parts of the event data within the data stream and how much information is within each event, is constructed in the PRUDE filter program, which includes pointers locating information in the body of the event from individual DYC's, denoting which hardware triggers fired, etc. If the topology of the event matches certain software triggers, it is further analyzed in the PRUDE filter program (Please Remove UnwanteD Events), and written to the appropriate buffer, which is most often a tape drive for storage. Just as the number of gates sent out to the FERA's by the triggers are prescaled, so too are the number of events that get written to different buffers of the Challenge. Thus not every event that passes one of the internal software triggers in PRUDE gets written to tape. However, a certain percentage of events are automatically written to buffers for calibration purposes via an autopass trigger regardless of whether it passes any of software triggers included in PRUDE. Rebuilt events from PRUDE may either be sent to a process called DOT (Data On Tape) to write the event to tape or to a process named HOIST, which serves the event display and the E835 Consumer program used for on-line monitoring.

One may change the many variables associated with each software and hardware trigger via a graphical interface built from the Tcl/Tk language[58]. For instance, one may want to activate or deactivate certain triggers before the run starts, change how often data passing a certain trigger gets written to a buffer (prescaling), or assign different buffers to the various triggers. Furthermore, one may want to create a new trigger, rename a trigger, view all the trigger bits making up each software trigger, or change the priorities of all the hardware triggers.

4.3. DART Software

The DART software architecture (Figure 4.3), of which the PRUDE filter is the central member, is highly functional and flexible. A client-server protocol over TCP/IP sockets forms the basic structure. Buffers are managed via a service provider/requestor algorithm, and the same application may be a provider in one circumstance and a requestor in another. Applications are registered with one or more groups, and different applications communicate through run control via group multicasting[59]. In other words, Run Control will send messages to all the members of a registered group.

The data acquisition is controlled via a single program called the *Operator Control Program* (OCP) through a user interface, which may occur via a command line or a graphical user interface (GUI). E835 uses a GUI to update the configuration database and start applications over several different nodes (i.e. several different SGI machines). DART provides a library of functions to allow applications to connect to servers, register to groups, and perform any kind of transaction. OCP is written in Tcl/Tk, which allows one to bind commands to buttons and exchange parameters in the database via entry boxes that one can type in.

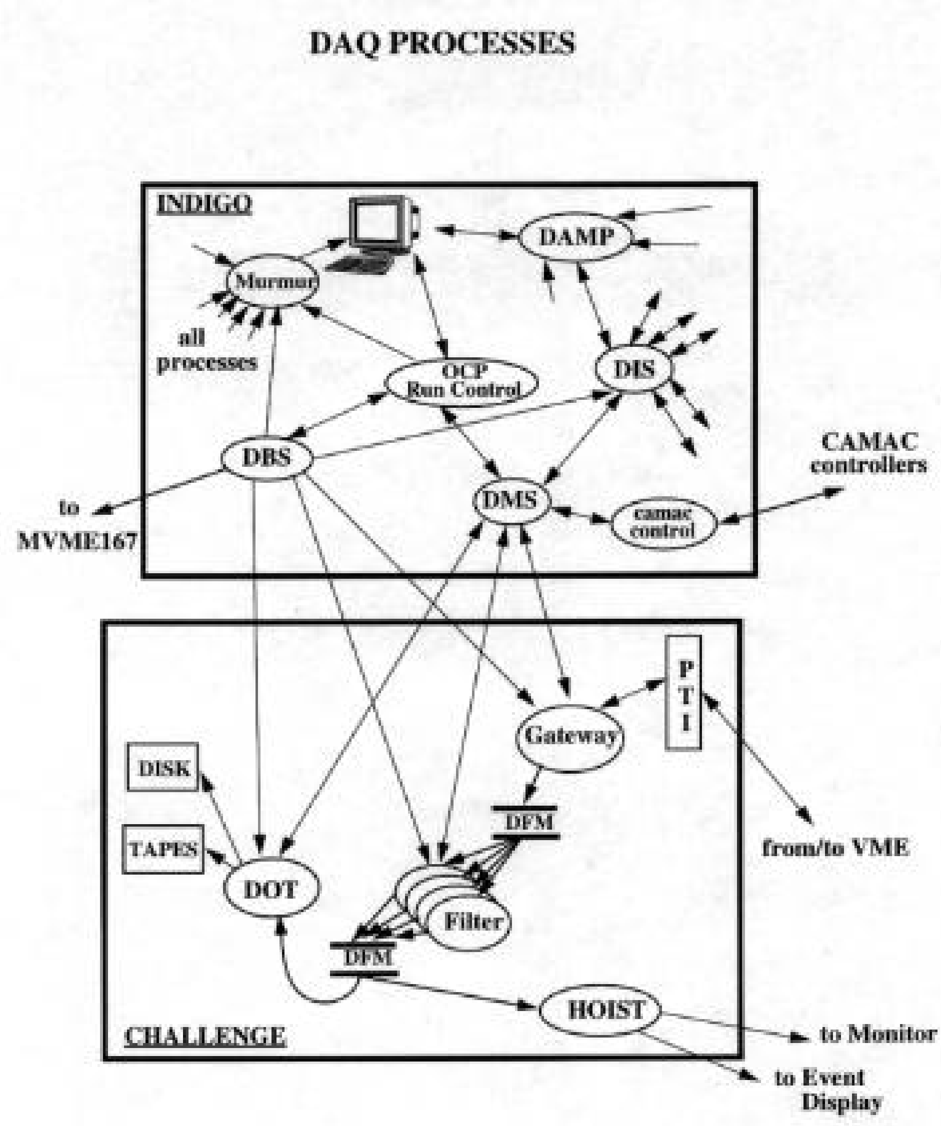

Figure 4.3 : Active DART software processes for the DAQ [60].

There are several servers that run while data acquisition is taking place. The two primary applications are the filter, which builds events from buffers passed to it via 3 different data streams, and the logger, which writes events that have passed certain hardware or software trigger requirements to tape or disk, as shown in Figure 4.4.

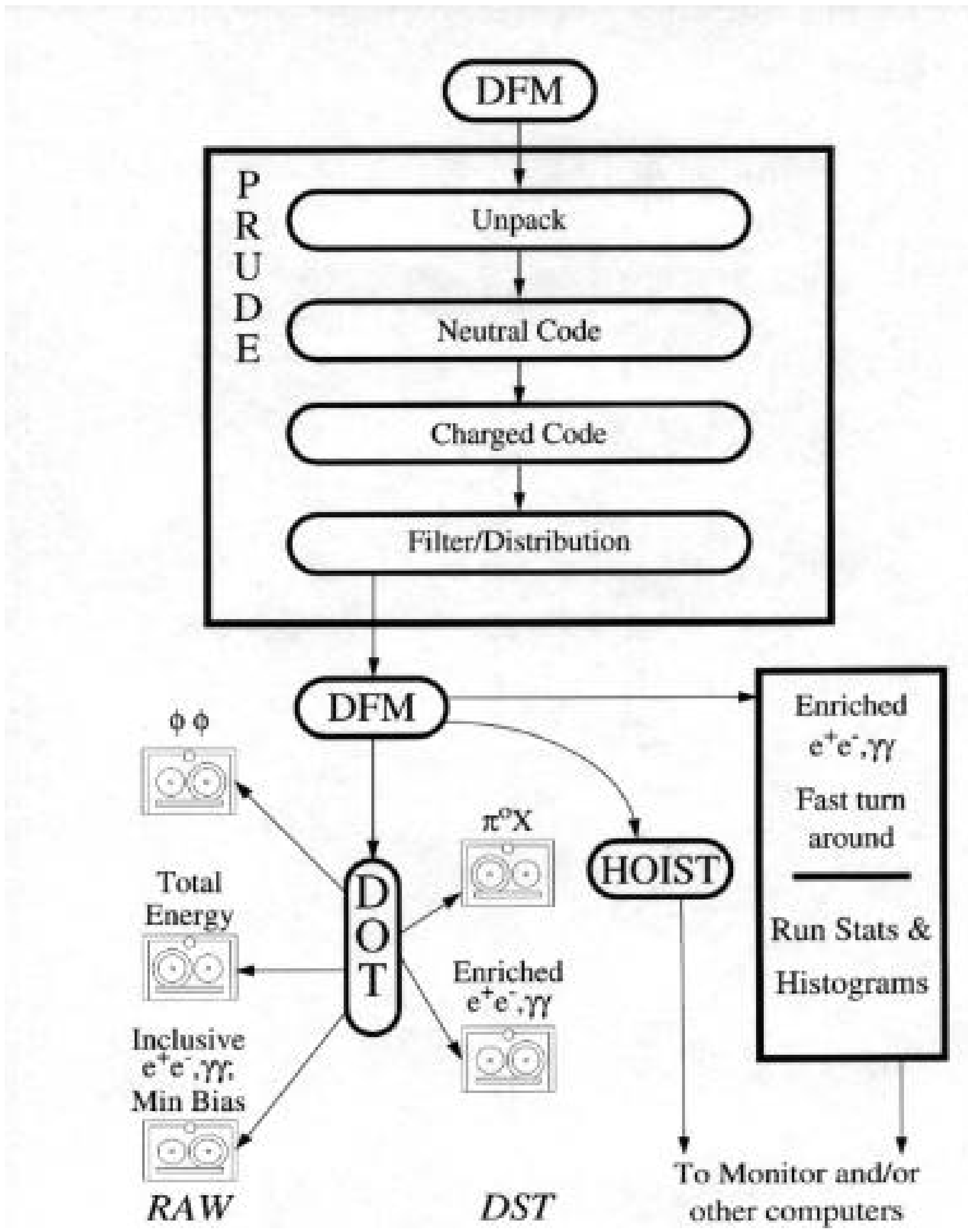

Figure 4.4 : The filter PRUDE and the logger [58].

However, in such a dynamic environment, buffers must be properly scheduled and organized to rebuild an event successfully, and all applications must be able to report errors and know where to look for buffers and/or messages. Furthermore, one wants to be able to monitor various rates and statistics for different processes. The *dis* framework (distributed information services) was designed by DART to perform these functions. A database of parameter values (numbers or strings) for each application is stored for a particular configuration (i.e. pure calibration, the taking of pedestals, and general data acquisition), and these statistics may be accessed during a run for monitoring purposes.

A server called *dfm* (data flow management) distributes buffers among various processes. Requestors queue buffers to different service providers that will use them. *dfm* schedules buffers via a list of buffer descriptors, which provide information about the buffer locations.

Messages from one application are multicast to all other processes of the same registered group via *dms* (distributed multicasting services). The advantages of *dms* are that messages can be distributed to applications on several different nodes as long as they are registered to the same group, commands are executed across the whole group, and members may join or leave groups dynamically. The application sending the message will wait for a reply from all members of the group before continuing.

DAQ (data acquisition) statistics may be monitored via *damp*, the Data Monitoring Program. *damp* provides a GUI builder (Graphical User Interface) to dynamically change the presentation of statistics (logging rates, percentage of tape used, buffer information) using various strip charts, bar graphs, etc. *damp* is closely related to the *dis* server in the manner in which it accesses parameters.

A certain percentage of events are “hoisted” to the on-line monitoring program called Consumer. Events are funnelled to various histograms, which may be dynamically viewed on-line by running Histoscope. In Histoscope, one can also dynamically adjust the binning of the histograms by clicking and dragging an axis with the mouse.

4.4. Dead Time

There are two dead times for the experiment[61]. One of these is caused by the Data Acquisition System itself. While the DAQ is processing an event, all triggers are vetoed, and therefore all events occurring during this “dead time” are lost. Run at a constant rate of $2.5 * 10^{31}$ nb^{-1} , the experiment was dead 2% of the time due to the DAQ (i.e. number of triggers lost compared with total number of triggers).

The second type of dead time is caused by the detectors themselves. Due to the high event rate, a veto counter may incur an accidental hit. If a trigger was formed but this veto counter was on, the event was disregarded. A study was done by looking at the veto counters in randomly opened gates during data taking, and it was determined that this contributed a dead time of about 10% to the experiment.

4.5. Operations

During 8-hour shifts around the clock the data acquisition system, various gas bottles used by the inner detectors, the status of the hydrogen gas jet, and the status of the detector elements are monitored. The E835 Monitor program looks after the DAQ and the various detector elements, a MacIntosh serves to monitor the gas jet, and gas levels are periodically examined manually.

Several programs run during each shift are necessary for the proper operation of the experiment. One of the programs critical to the basic operation of the experiment is the High Voltage Program[62,63]. E835 uses six LeCroy 1440 high voltage mainframes to power the photomultiplier tubes used by the hodoscopes, Cerenkov counters, central calorimeter, and forward calorimeter.

The E760 version of the program provided a text-based interface that wrote commands to, and read responses from, the RS232 port of the Controller for each 1440 mainframe. Since E835 does not use any VAX-based computers, the program's I/O routines had to be rewritten in order to communicate with the 1440 via a serial port on an SGI running UNIX. UNIX does not provide a set of easy-to-use I/O routines such as the SYS$QIOW routine that VAX does.

A High Voltage Monitor Program[64] is also available to alert Run Control of any changes in the status of the LeCroy mainframes during a shift. The program polls the mainframes every so often and reports an error if a mainframe does not respond to a command, the response is abnormal, or the voltage read back is outside a certain window. Both the High Voltage Control Program and the High Voltage Monitor are designed to lock out the SGI serial port so only one user may have access to the high voltage mainframes at any one time.

Pedestals for the experiment may be taken at any time by running the data acquisition system software in the pedestal configuration. Usually 10-20,000 events generated by a pulser are taken, and the pedestal program[65] then evaluates means and widths for every FERA ADC channel in the experiment. One problem that the pedestal program must address is spurious large hits due to large cosmic rays that may pass through the detector at the time the gate is opened. One or two such hits per channel may dramatically impact the pedestal mean. To alleviate this problem, a subset of a couple hundred events are used to establish a mean plus 3 σ window (If σ is too small, then the default window is mean plus 30 ADC counts). If a pedestal is larger than this threshold, it is not used in the evaluation of the pedestal mean for that channel.

Once the requested number of events have been taken , the pedestal program opens an interface which reports how many and what kind of errors occurred, and logs error information for each channel to a file. The user may view the current pedestals, compare them with a different set of pedestals, or write the pedestals to a file or to the E835 database.

Pedestals and calibration constants for the experiment are stored in the E835 database by the mSQL database package. The package includes routines to write and read entries of the columns (or fields) of tables, which form the structure of the database. A product called *dbmsql* has been installed on both offline and online machines, so that run-time DAQ applications may access the database as well as offline analyses[66].